



UNITED STATES PATENT AND TRADEMARK OFFICE

89
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/875,245	06/05/2001	Ashvinkumar J. Sanghvi	MSI-701US	7289
22801	7590	04/21/2006	EXAMINER	
LEE & HAYES PLLC 421 W RIVERSIDE AVENUE SUITE 500 SPOKANE, WA 99201				PATEL, HARESH N
			ART UNIT	PAPER NUMBER
			2154	

DATE MAILED: 04/21/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/875,245	SANGHVI ET AL.	
	Examiner Haresh Patel	Art Unit 2154	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 07 February 2006.

2a) This action is FINAL. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-29 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-29 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 12/15/03, 11/17/03, 3/4/04, 3/14/05, 07/21/05

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____

5) Notice of Informal Patent Application (PTO-152)

6) Other: _____

DETAILED ACTION

1. Claims 1-29 are presented for examination.

Response to Arguments

2. Applicant's arguments with respect to claims 1-29 have been considered but are moot in view of the new ground(s) of rejection.
3. Applicant's arguments presented in the appeal brief, dated 2/7/2006, regarding the claimed subject matter of the claims is persuasive and, therefore, the finality of office action, dated 1/25/2005, is withdrawn and the prosecution is hereby reopened. However, upon further consideration of the available prior arts, the claimed subject matter is rejected with the new grounds of rejection.

Priority

4. Claims 1-29 do not benefit the provisional priority date as the effective filling date, because the provisional application does not contain all the limitations of the claimed invention, including the description of payload objects and use of event filters. For example, the provisional application does not contain limitations, “the second event filter being associated with an event consumer”, “the event consumer performs an action if the second event satisfies the filter criteria associated with the second event filter event”, “the second event filter has no knowledge of the first event”, “generating an event header”, “the event transformer operates independently of the event filters and independently of the event consumer”.

5. Claims 1-29 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-17 of copending Application No. 2005/0071849. Although the conflicting claims are not identical, they are not patentably distinct from each other because the limitations of the claims 1-29 are similar to claim 1-17 of the copending application. The claimed subject matter of the copending application does not specifically mention about event transforming. However, the concept of event transforming is well-known in the art for example Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32). With the Paquette's teachings it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Paquette with the claimed subject matter of the copending application to facilitate usage of event transforming because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

6. Claims 1-29 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-20 of copending Application No. 2005/0044554. Although the conflicting claims are not identical, they are not patentably distinct from each other because the limitations of the claims 1-29 are similar to claim 1-20 of the copending application. The claimed subject matter of the copending application does not specifically mention about event transforming. However, the concept of event transforming

is well-known in the art for example Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32). With the Paquette's teachings it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Paquette with the claimed subject matter of the copending application to facilitate usage of event transforming because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

7. Claims 1-29 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-25 of Hinson et al., U.S. Patent No. 6,748,455. Although the conflicting claims are not identical, they are not patentably distinct from each other because the limitations of the claims 1-29 are similar to claim 1-25 of the patent. The claimed subject matter of the patent does not specifically mention about event transforming. However, the concept of event transforming is well-known in the art for example Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32). With the Paquette's teachings it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Paquette with the claimed subject matter of the patent to facilitate usage of event transforming because the transforming would enhance converting one event into another event. The event transformer would support

implementing the converting. Another converted event would be available for the system for processing.

8. Claims 1-29 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-9 of Hinson et al., U.S. Patent No. 6,829,770. Although the conflicting claims are not identical, they are not patentably distinct from each other because the limitations of the claims 1-29 are similar to claim 1-9 of the patent. The claimed subject matter of the patent does not specifically mention about event transforming. However, the concept of event transforming is well-known in the art for example Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32). With the Paquette's teachings it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Paquette with the claimed subject matter of the patent to facilitate usage of event transforming because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

9. Claims 1-29 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-15 of Jakobson et al., U.S. Patent No. 6,766,368. Although the conflicting claims are not identical, they are not patentably distinct from each other because the limitations of the claims 1-29 are similar to claim 1-15 of the patent. The claimed subject matter of the patent does not specifically mention about event transforming. However,

the concept of event transforming is well-known in the art for example Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32). With the Paquette's teachings it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Paquette with the claimed subject matter of the patent to facilitate usage of event transforming because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

Claim Objections

10. Claims 2-9, 12-14, 23-29 is objected to because of the following informalities:

Claims 2-9, 12-14, 28 and 29 mentions, "A method as recited in claim", which should be --The method as recited in claim--.

Claims 17-21 mentions, "An apparatus as recited in claim", which should be --The apparatus as recited in claim--.

Claims 23-27 mentions, "One or computer-readable media as recited in claim", which should be --The one or computer-readable media as recited in claim--.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hereinafter O'Brien in view of Paquette et al., 2001/0049086 (Hereafter Paquette).

13. As per claims 1, 10, 11, 15, 16 and 22, O'Brien teaches the following:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., an event generated by SNMP-capable agent is received by the SNMP Traps network event, col. 2, line 60 - col. 7, line 65), the first event filter having an associated filter criteria (e.g. an agent having a vendor-specific MIB to interpret specific SNMP traps, col. 2, line 60 – col. 6, line65),

applying the filter criteria associated with the first event filter to the first event (e.g., a network event 7 could be either a standard SNMP trap 31 or a specialized trap or notification, such as a certogram 30. In addition, three types of event filters are used, Visual Basic scripts, frequency thresholds, and adding custom values to the event or action based on SNMP values before the trap is forwarded to an action 32. Other forms of network events, actions and event filters are feasible as is known in the art, col. 5, lines 6 – 50),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., mapping of network events 7 to actions 32, col. 5, line 6 – col. 6, line 65); and

communicating the second event to a second event filter having an associated filter criteria (e.g., the arbiter 8 manages the receipt, analysis, and processing of network events 7 based on user defined action sets 39. Each action set 39 defines mappings between actions 32 and event filters 40. Actions 32 are performed by actors 9 and are described below. Event filters 40 are actions which are performed after the receipt of a network event 7 but before any corresponding actions 32 are executed, col. 2, line 61, col. 6, line 65),

the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the second event filter (e.g., the actors 9 take actions in response to instructions from the arbiter 8. In the described embodiment, six types of actors are supported, a help desk 27, firewall 28, and methods for broadcasting a network message 33, sending an electronic mail message (email) 34, sending an alphanumeric page 35, and running a script 36. Other types of actors and actions are feasible. The methods could be performed by the same system as the manager 25 or by another remote system depending upon the definition of the action, col. 2, line 61, col. 6, line 65),

receiving a first event having a first format (e.g., an agent having a vendor-specific MIB supplied with their hardware and other types of sensors may have unique mapping key formats, col. 2, line 61, col. 6, line 65),

processing the first event into a second event having a second format (e.g., format of the events handled by the actions, figure 2, col. 2, line 61, col. 6, line 65),

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., tags used in figure 4, is a data structure showing a binary large object (BLOB) 60 storing an action set 32 for use in the manager 25, Using the BLOB 60, an action set 32 can embed one or more actions 32 and can reference one or more event filters 40. An action set 32 forms an association with one or more network events 7 through scope, as the event mappings are stored in the database 26, including certogram mappings 48 and SNMP mappings 49, within an instance of an action set 32. Internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., The arbiter 8 manages the receipt, analysis, and processing of network events 7 based on user defined action sets 39. Each action set 39 defines mappings between actions 32 and event filters 40, col. 5, lines 35 – 60),

an event processor to receive a first event and processing the first event into a second event (e.g., network events receiving events from the agents and then generating events for the manager, col. 2, line 60 - col. 7, line 65), the second event having a standard data format regardless of the first event data format (e.g., action set containing user defined actions, col. 2, line 60 - col. 7, line 65),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., event filters and their associated action sets, figure 2),

a plurality of event consumers coupled to the plurality of event filters (e.g., actions coupled with the manager and event filters, figure 2, col. 2, line 60 – col. 7, line 65) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., actions 32, to create actions based on the events, figure 2, col. 2, line 60 - col. 7, line 65),

receive a first event having a first data format (e.g., receiving an event from the agents having hardware dependent format, an agent have a copy of a vendor-specific MIB supplied with their hardware, Other types of sensors may have unique mapping key formats, col. 2, line 61, col. 6, line 65),

filter the first event using a first filter criteria (e.g., an agent filtering using a vendor-specific MIB, col. 2, line 60 – col. 6, line65),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., mapping of network events 7 to actions 32, col. 5, line 6 – col. 6, line 65), wherein the second data format includes an event header having a plurality of parameters (e.g., internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col.

10, line 43), and an event payload having a plurality of payload objects (e.g., manager and actions sets, event filters and actions, col. 2, line 60 - col. 7, line 65); and

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., to send an e-mail by actions after manger receives events from network events, col. 2, line 60 - col. 7, line 65).

However, O'Brien does not specifically mention about transforming and usage of event transformer.

Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of O'Brien with the teachings of Paquette in order to facilitate transforming and usage of event transformer because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

14. As per claims 2-9, 12-14, 17-21 and 23-27, O'Brien and Paquette teach the claimed limitations rejected above. O'Brien also discloses the following:

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., a data structure showing a binary large object (BLOB) 60 storing an action set 32 for use in the manager 25 of the system of FIG. 1. Using the BLOB 60, an action set 32 can embed one or more actions 32 and can reference one or more event filters 40. An action set 32 forms an association with one or more

Art Unit: 2154

network events 7 through scope, as the event mappings are stored in the database 26, including certogram mappings 48 and SNMP mappings 49, within an instance of an action set 32.

Internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., manager and actions sets, event filters and actions, action blocks determining and sending out user defined events, col. 2, line 60 - col. 7, line 65),

the second event filter has no knowledge of the first event (e.g., independent network events, manager and event filters, col. 2, line 60 - col. 7, line 65),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., events sent to the event filters by the network elements, col. 2, line 60 - col. 7, line 65), each of the plurality of event filters having an associated filter criteria (e.g., event filters must be configured in a manner analogous to actions before being assigned to an action set. Configured filters are stored as BLOBS 60 in a global table, col. 10, lines 8 – 42),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., event filters for different actions, figure 2), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with

the corresponding event filter (e.g., different actions blocks 32 performing different actions, figure 2).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., usage of storage manager and database, figure 3).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., broadcast network message action, figure 3),

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., send e-mail action, figure 3),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., a data structure showing a binary large object (BLOB) 60 storing an action set 32 for use in the manager 25, Using the BLOB 60, an action set 32 can embed one or more actions 32 and can reference one or more event filters 40. An action set 32 forms an association with one or more network events 7 through scope, as the event mappings are stored in the database 26, including certogram mappings 48 and SNMP mappings 49, within an instance of an action set 32. Internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-

c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., In response to the receipt of the network event notification, the manager 25 determines the action set 39 to which the network event 7 is associated using a stored set of event mappings. Each of these action sets 39 group one or more actions 32 and one or more optional event filters 40. If appropriate, the manager 25 causes an actor 9 to perform some task by dispatching an appropriate action 32, col. 3, line 61, col. 10, line 41),

the event processor operates independently of the event filters and independently of the event consumers (e.g., manager independent of the event filters and the actions, figure 2),

the plurality of payload objects in the event payload are used by an event consumer (e.g., action blocks, figure 2) that receives the second event to identify an action to perform in response to the second event (e.g., action blocks to perform actions, to send e-mail, user notification etc., figure 2),

the event action handler performs at least one action in response to the second event (e.g., action blocks to perform actions, to send e-mail, user notification etc., figure 2).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figure 2, col. 3, line 61, col. 10, line 41, Also, e.g., paragraphs 29-32 of Paquette).

15. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wold et. al. 5,724,589 (Hereinafter Wold) in view of Paquette et al., 2001/0049086 (Hereafter Paquette).

16. As per claims 1-29, Wold teaches the following:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., event sink, col. 9, line 36 – col. 10, line 21), the first event filter having an associated filter criteria (e.g., even sink handling event, col. 9, line 36 – col. 10, line 21)

applying the filter criteria associated with the first event filter to the first event (e.g., even sink handling event, col. 9, line 36 – col. 10, line 21),

processing the first event into a second event (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21)

communicating the second event to a second event filter having an associated filter criteria (e.g., even sink handling event, col. 9, line 36 – col. 10, line 21)

the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the

second event filter (e.g., any object oriented event that can be handled by the event filter in the event chain, (e.g., col. 9, line 36 – col. 10, line 21),

receiving a first event having a first format (e.g., abstract),

processing the first event into a second event having a second format (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21),

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., usage of object oriented terms and programming, macro with parameters, col. 11, line 15 – col. 23, line 49),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., sink object to handle the event, col. 9, line 36 – col. 10, line 21, abstract),

an event processor to receive a first event and process the first event into a second event (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21),

a plurality of event consumers coupled to the plurality of event filters (e.g., event sink of the event chain filters) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., action performed by the event sink in an event chain, col. 9, line 36 – col. 10, line 21),

receive a first event having a first data format, filter the first event using a first filter criteria, process the first event into a second event having a second data format if the first event satisfies the first filter criteria, communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., Despite the absence of any C++ language support for events, the present invention provides a type-safe "wiring" mechanism--one using standard C++ to dispatch an event, raised by one object (the "event source"), to a method of another object (the "event sink"), with the requirement that the event source does not need to know the class of the event sink. As a result, the system allows developers to create C++ software components which can be connected together without the components having to know anything about the makeup of the component to which it is connected. Thus, developers can create pre-packaged, re-usable software components which can simply be "plugged into" a design--all accomplished within the confines of the standard C++ programming language, abstract).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., col. 7, line 1 - col. 10, line 43),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., col. 2, line 60 - col. 7, line 65),

the second event filter has no knowledge of the first event (e.g., col. 2, line 60 - col. 7, line 65),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., col. 2, line 60 - col. 7, line

Art Unit: 2154

65), each of the plurality of event filters having an associated filter criteria (e.g., col. 10, lines 8 – 42),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figure 2, col. 3, line 61 - col. 10, line 41), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figure 2, col. 3, line 61 - col. 10, line 41).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figure 3, col. 3, line 61 - col. 10, line 41),

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figure 3, col. 3, line 61 - col. 10, line 41),

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figure 3, col. 3, line 61 - col. 10, line 41),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., col. 7, line 1, col. 10, line 43),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and

communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., col. 3, line 61 - col. 10, line 41),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figure 2, col. 3, line 61 - col. 10, line 41),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figure 2, col. 3, line 61 - col. 10, line 41) that receives the second event to identify an action to perform in response to the second event (e.g., figure 2, col. 3, line 61 - col. 10, line 41),

the event action handler performs at least one action in response to the second event (e.g., figure 2, figure 2, col. 3, line 61 - col. 10, line 41).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figure 2, col. 3, line 61, col. 10, line 41).

However, Wold does not specifically mention about transforming and usage of event transformer.

Paquette discloses the well-known concept of transforming and usage of event transformer (e.g., paragraphs 29-32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Wold with the teachings of Paquette in order to facilitate transforming and usage of event transformer because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

17. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hereinafter O'Brien in view of Bonnell et al., 5,655,081 (Hereafter Bonnell).

18. As per claims 1, 10, 11, 15, 16 and 22, O'Brien teaches the following:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., an event generated by SNMP-capable agent is received by the SNMP Traps network event, col. 2, line 60 - col. 7, line 65), the first event filter having an associated filter criteria (e.g. an agent having a vendor-specific MIB to interpret specific SNMP traps, col. 2, line 60 – col. 6, line65),

applying the filter criteria associated with the first event filter to the first event (e.g., a network event 7 could be either a standard SNMP trap 31 or a specialized trap or notification, such as a certogram 30. In addition, three types of event filters are used, Visual Basic scripts, frequency thresholds, and adding custom values to the event or action based on SNMP values before the trap is forwarded to an action 32. Other forms of network events, actions and event filters are feasible as is known in the art, col. 5, lines 6 – 50),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., mapping of network events 7 to actions 32, col. 5, line 6 – col. 6, line 65); and

Art Unit: 2154

communicating the second event to a second event filter having an associated filter criteria (e.g., the arbiter 8 manages the receipt, analysis, and processing of network events 7 based on user defined action sets 39. Each action set 39 defines mappings between actions 32 and event filters 40. Actions 32 are performed by actors 9 and are described below. Event filters 40 are actions which are performed after the receipt of a network event 7 but before any corresponding actions 32 are executed, col. 2, line 61, col. 6, line 65),

the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the second event filter (e.g., the actors 9 take actions in response to instructions from the arbiter 8. In the described embodiment, six types of actors are supported, a help desk 27, firewall 28, and methods for broadcasting a network message 33, sending an electronic mail message (email) 34, sending an alphanumeric page 35, and running a script 36. Other types of actors and actions are feasible. The methods could be performed by the same system as the manager 25 or by another remote system depending upon the definition of the action, col. 2, line 61, col. 6, line 65),

receiving a first event having a first format (e.g., an agent having a vendor-specific MIB supplied with their hardware and other types of sensors may have unique mapping key formats, col. 2, line 61, col. 6, line 65),

processing the first event into a second event having a second format (e.g., format of the events handled by the actions, figure 2, col. 2, line 61, col. 6, line 65),

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., tags used in figure 4, is a data structure showing a binary large object (BLOB) 60 storing an action set 32 for use in the manager 25,

Using the BLOB 60, an action set 32 can embed one or more actions 32 and can reference one or more event filters 40. An action set 32 forms an association with one or more network events 7 through scope, as the event mappings are stored in the database 26, including certogram mappings 48 and SNMP mappings 49, within an instance of an action set 32. Internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., The arbiter 8 manages the receipt, analysis, and processing of network events 7 based on user defined action sets 39. Each action set 39 defines mappings between actions 32 and event filters 40, col. 5, lines 35 – 60),

an event processor to receive a first event and processing the first event into a second event (e.g., network events receiving events from the agents and then generating events for the manager, col. 2, line 60 - col. 7, line 65), the second event having a standard data format regardless of the first event data format (e.g., action set containing user defined actions, col. 2, line 60 - col. 7, line 65),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., event filters and their associated action sets, figure 2),

a plurality of event consumers coupled to the plurality of event filters (e.g., actions coupled with the manager and event filters, figure 2, col. 2, line 60 – col. 7, line 65) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., actions 32, to create actions based on the events, figure 2, col. 2, line 60 - col. 7, line 65),

receive a first event having a first data format (e.g., receiving an event from the agents having hardware dependent format, an agent have a copy of a vendor-specific MIB supplied with their hardware, Other types of sensors may have unique mapping key formats, col. 2, line 61, col. 6, line 65),

filter the first event using a first filter criteria (e.g., an agent filtering using a vendor-specific MIB, col. 2, line 60 – col. 6, line65),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., mapping of network events 7 to actions 32, col. 5, line 6 – col. 6, line 65), wherein the second data format includes an event header having a plurality of parameters (e.g., internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43), and an event payload having a plurality of payload objects (e.g., manager and actions sets, event filters and actions, col. 2, line 60 - col. 7, line 65); and

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., to send an e-mail by actions after manager receives events from network events, col. 2, line 60 - col. 7, line 65).

However, O'Brien does not specifically mention about transforming and usage of event transformer.

Bonnell discloses the well-known concept of transforming and usage of event transformer (e.g., figures 15, 17, 23, 25 and related description regarding event filters in the chain of filters, col., 10, line 53 – col., 11, line 48, col., 13, line 5 – col., 14, line 38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of O'Brien with the teachings of Bonnell in order to facilitate transforming and usage of event transformer because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

19. As per claims 2-9, 12-14, 17-21 and 23-27, O'Brien and Bonnell teach the claimed limitations rejected above. O'Brien also discloses the following:

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., a data structure showing a binary large object (BLOB) 60 storing an action set 32 for use in the manager 25 of the system of FIG. 1. Using the BLOB 60, an action set 32 can embed one or more actions 32 and can reference one or more event filters 40. An action set 32 forms an association with one or more network events 7 through scope, as the event mappings are stored in the database 26, including

certogram mappings 48 and SNMP mappings 49, within an instance of an action set 32. Internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., manager and actions sets, event filters and actions, action blocks determining and sending out user defined events, col. 2, line 60 - col. 7, line 65),

the second event filter has no knowledge of the first event (e.g., independent network events, manager and event filters, col. 2, line 60 - col. 7, line 65),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., events sent to the event filters by the network elements, col. 2, line 60 - col. 7, line 65), each of the plurality of event filters having an associated filter criteria (e.g., event filters must be configured in a manner analogous to actions before being assigned to an action set. Configured filters are stored as BLOBs 60 in a global table, col. 10, lines 8 – 42),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., event filters for different actions, figure 2), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with

Art Unit: 2154

the corresponding event filter (e.g., different actions blocks 32 performing different actions, figure 2).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., usage of storage manager and database, figure 3).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., broadcast network message action, figure 3),

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., send e-mail action, figure 3),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., a data structure showing a binary large object (BLOB) 60 storing an action set 32 for use in the manager 25, Using the BLOB 60, an action set 32 can embed one or more actions 32 and can reference one or more event filters 40. An action set 32 forms an association with one or more network events 7 through scope, as the event mappings are stored in the database 26, including certogram mappings 48 and SNMP mappings 49, within an instance of an action set 32. Internally, each action set 32 stores the action set configuration 61, including any variables used by the action set 62, and one or more streams 63a-

c within which are stored the specifications for actions 32. Each stream 63a contains a stream identifier 64, Globally Unique Identifier (GUID) 65, name 66, configuration 67, and any variables 68 used by the action. Also, if used, any filter identifiers 69 are also stored in the BLOB 60, col. 7, line 1, col. 10, line 43),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., In response to the receipt of the network event notification, the manager 25 determines the action set 39 to which the network event 7 is associated using a stored set of event mappings. Each of these action sets 39 group one or more actions 32 and one or more optional event filters 40. If appropriate, the manager 25 causes an actor 9 to perform some task by dispatching an appropriate action 32, col. 3, line 61, col. 10, line 41),

the event processor operates independently of the event filters and independently of the event consumers (e.g., manager independent of the event filters and the actions, figure 2),

the plurality of payload objects in the event payload are used by an event consumer (e.g., action blocks, figure 2) that receives the second event to identify an action to perform in response to the second event (e.g., action blocks to perform actions, to send e-mail, user notification etc., figure 2),

the event action handler performs at least one action in response to the second event (e.g., action blocks to perform actions, to send e-mail, user notification etc., figure 2).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figure 2, col. 3, line 61, col. 10, line 41, Also, e.g., figure 23 of Bonnell).

20. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wold et. al. 5,724,589 (Hereinafter Wold) in view of Bonnell.

21. As per claims 1-29, Wold teaches the following:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., event sink, col. 9, line 36 – col. 10, line 21), the first event filter having an associated filter criteria (e.g., even sink handling event, col. 9, line 36 – col. 10, line 21)

applying the filter criteria associated with the first event filter to the first event (e.g., even sink handling event, col. 9, line 36 – col. 10, line 21),

processing the first event into a second event (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21)

communicating the second event to a second event filter having an associated filter criteria (e.g., even sink handling event, col. 9, line 36 – col. 10, line 21)

the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the

Art Unit: 2154

second event filter (e.g., any object oriented event that can be handled by the event filter in the event chain, (e.g., col. 9, line 36 – col. 10, line 21),

receiving a first event having a first format (e.g., abstract),

processing the first event into a second event having a second format (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21),

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., usage of object oriented terms and programming, macro with parameters, col. 11, line 15 – col. 23, line 49),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., sink object to handle the event, col. 9, line 36 – col. 10, line 21, abstract),

an event processor to receive a first event and process the first event into a second event (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., transfer from one event to another event in an event chain, col. 9, line 36 – col. 10, line 21),

a plurality of event consumers coupled to the plurality of event filters (e.g., event sink of the event chain filters) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., action performed by the event sink in an event chain, col. 9, line 36 – col. 10, line 21),

receive a first event having a first data format, filter the first event using a first filter criteria, process the first event into a second event having a second data format if the first event satisfies the first filter criteria, communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., Despite the absence of any C++ language support for events, the present invention provides a type-safe "wiring" mechanism--one using standard C++ to dispatch an event, raised by one object (the "event source"), to a method of another object (the "event sink"), with the requirement that the event source does not need to know the class of the event sink. As a result, the system allows developers to create C++ software components which can be connected together without the components having to know anything about the makeup of the component to which it is connected. Thus, developers can create pre-packaged, re-usable software components which can simply be "plugged into" a design--all accomplished within the confines of the standard C++ programming language, abstract).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., col. 7, line 1 - col. 10, line 43),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., col. 2, line 60 - col. 7, line 65),

the second event filter has no knowledge of the first event (e.g., col. 2, line 60 - col. 7, line 65),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., col. 2, line 60 - col. 7, line

Art Unit: 2154

65), each of the plurality of event filters having an associated filter criteria (e.g., col. 10, lines 8 – 42),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figure 2, col. 3, line 61 - col. 10, line 41), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figure 2, col. 3, line 61 - col. 10, line 41).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figure 3, col. 3, line 61 - col. 10, line 41),

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figure 3, col. 3, line 61 - col. 10, line 41),

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figure 3, col. 3, line 61 - col. 10, line 41),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., col. 7, line 1, col. 10, line 43),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and

communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., col. 3, line 61 - col. 10, line 41),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figure 2, col. 3, line 61 - col. 10, line 41),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figure 2, col. 3, line 61 - col. 10, line 41) that receives the second event to identify an action to perform in response to the second event (e.g., figure 2, col. 3, line 61 - col. 10, line 41),

the event action handler performs at least one action in response to the second event (e.g., figure 2, figure 2, col. 3, line 61 - col. 10, line 41).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figure 2, col. 3, line 61, col. 10, line 41).

However, Wold does not specifically mention about transforming and usage of event transformer.

Bonnell discloses the well-known concept of transforming and usage of event transformer (e.g., figures 15, 17, 23, 25 and related description regarding event filters in the chain of filters, col., 10, line 53 – col., 11, line 48, col., 13, line 5 – col., 14, line 38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Wold with the teachings of Bonnell in order to facilitate transforming and usage of event transformer because the transforming would enhance converting one event into another event. The event transformer would support implementing the converting. Another converted event would be available for the system for processing.

Claim Rejections - 35 USC § 102

22. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

23. Claims 1-29, are rejected under 35 U.S.C. 102(e) as being anticipated by Hinson et al.

6,748,455 (Hereinafter Hinson1).

24. As per claims 1-29, Hinson1 teaches a method comprising:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., figures 3 –35 and related description), the first event filter having an associated filter criteria (e.g., figures 3 –35 and its related description),

applying the filter criteria associated with the first event filter to the first event (e.g., figures 3 –35 and its related description),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., figures 3 –35 and its related description); and

communicating the second event to a second event filter having an associated filter criteria (e.g., figures 3 –35 and its related description),
the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the second event filter (e.g., figures 3 –35 and its related description),
receiving a first event having a first format (e.g., figures 3 –35 and its related description),
processing the first event into a second event having a second format (e.g., figures 3 –35 and its related description),
generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., figures 3 –35 and its related description),
generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),
an event processor to receive a first event and processing the first event into a second event (e.g., figures 3 –35 and its related description), the second event having a standard data format regardless of the first event data format (e.g., figures 3 –35 and its related description),
a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., figures 3 –35 and its related description),
a plurality of event consumers coupled to the plurality of event filters (e.g., figures 3 –35 and its related description) and the event consumers to perform an action if the second event

Art Unit: 2154

satisfies the filter criteria applied by the event filters (e.g., figures 3 –35 and its related description),

receive a first event having a first data format (e.g., figures 3 –35 and its related description),

filter the first event using a first filter criteria (e.g., figures 3 –35 and its related description),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description), wherein the second data format includes an event header having a plurality of parameters (e.g., figures 3 –35 and its related description), and an event payload having a plurality of payload objects (e.g., figures 3 –35 and its related description); and

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description).

the well-known concept of transforming and usage of event transformer (e.g., figures 3 –35 and its related description).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., figures 3 –35 and its related description),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),

the second event filter has no knowledge of the first event (e.g., figures 3 –35 and its related description),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., figures 3 –35 and its related description), each of the plurality of event filters having an associated filter criteria (e.g., figures 3 –35 and its related description),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figures 3 –35 and its related description), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figures 3 –35 and its related description)

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figures 3 –35 and its related description),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the

Art Unit: 2154

plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., figures 3 –35 and its related description),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., figures 3 –35 and its related description),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figures 3 –35 and its related description),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figures 3 –35 and its related description) that receives the second event to identify an action to perform in response to the second event (e.g., figures 3 –35 and its related description),

the event action handler performs at least one action in response to the second event (e.g., figures 3 –35 and its related description).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figures 3 –35 and its related description).

25. Claims 1-29, are rejected under 35 U.S.C. 102(e) as being anticipated by Hinson et al. 6,829,770 (Hereinafter Hinson2).

26. As per claims 1-29, Hinson2 teaches a method comprising:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., figures 3 –35 and related description),
the first event filter having an associated filter criteria (e.g., figures 3 –35 and its related
description),

applying the filter criteria associated with the first event filter to the first event (e.g.,
figures 3 –35 and its related description),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., figures 3 –35 and its related
description); and

communicating the second event to a second event filter having an associated filter
criteria (e.g., figures 3 –35 and its related description),

the second event filter being associated with an event consumer wherein the event
consumer performs an action if the second event satisfies the filter criteria associated with the
second event filter (e.g., figures 3 –35 and its related description),

receiving a first event having a first format (e.g., figures 3 –35 and its related description
(),

processing the first event into a second event having a second format (e.g., figures 3 –35
and its related description),

generating an event header having a plurality of parameters, wherein the plurality of
parameters are arranged in a standard data format (e.g., figures 3 –35 and its related description),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 – 35 and its related description),

an event processor to receive a first event and processing the first event into a second event (e.g., figures 3 –35 and its related description), the second event having a standard data format regardless of the first event data format (e.g., figures 3 –35 and its related description),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., figures 3 –35 and its related description),

a plurality of event consumers coupled to the plurality of event filters (e.g., figures 3 –35 and its related description) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., figures 3 –35 and its related description),

receive a first event having a first data format (e.g., figures 3 –35 and its related description),

filter the first event using a first filter criteria (e.g., figures 3 –35 and its related description),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description), wherein the second data format includes an event header having a plurality of parameters (e.g., figures 3 –35 and its related description), and an event payload having a plurality of payload objects (e.g., figures 3 –35 and its related description); and

Art Unit: 2154

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description).

the well-known concept of transforming and usage of event transformer (e.g., figures 3 –35 and its related description).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., figures 3 –35 and its related description),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),

the second event filter has no knowledge of the first event (e.g., figures 3 –35 and its related description),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., figures 3 –35 and its related description), each of the plurality of event filters having an associated filter criteria (e.g., figures 3 –35 and its related description),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figures 3 –35 and its related description), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figures 3 –35 and its related description).

Art Unit: 2154

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figures 3 –35 and its related description)

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figures 3 –35 and its related description),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., figures 3 –35 and its related description),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., figures 3 –35 and its related description),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figures 3 –35 and its related description),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figures 3 –35 and its related description) that receives the second event to identify an action to perform in response to the second event (e.g., figures 3 –35 and its related description),

· the event action handler performs at least one action in response to the second event (e.g., figures 3 –35 and its related description).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figures 3 –35 and its related description).

27. Claims 1-29, are rejected under 35 U.S.C. 102(e) as being anticipated by Hinson et al. 2005/0071849 (Hereinafter Hinson3).

28. As per claims 1-29, Hinson3 teaches a method comprising:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., figures 3 –35 and related description), the first event filter having an associated filter criteria (e.g., figures 3 –35 and its related description),

applying the filter criteria associated with the first event filter to the first event (e.g., figures 3 –35 and its related description),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., figures 3 –35 and its related description); and

communicating the second event to a second event filter having an associated filter criteria (e.g., figures 3 –35 and its related description),

the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the second event filter (e.g., figures 3 –35 and its related description),

receiving a first event having a first format (e.g., figures 3 –35 and its related description),

processing the first event into a second event having a second format (e.g., figures 3 –35 and its related description),

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., figures 3 –35 and its related description),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),

an event processor to receive a first event and processing the first event into a second event (e.g., figures 3 –35 and its related description), the second event having a standard data format regardless of the first event data format (e.g., figures 3 –35 and its related description),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., figures 3 –35 and its related description),

Art Unit: 2154

a plurality of event consumers coupled to the plurality of event filters (e.g., figures 3 –35 and its related description) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., figures 3 –35 and its related description),

receive a first event having a first data format (e.g., figures 3 –35 and its related description),

filter the first event using a first filter criteria (e.g., figures 3 –35 and its related description),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description), wherein the second data format includes an event header having a plurality of parameters (e.g., figures 3 –35 and its related description), and an event payload having a plurality of payload objects (e.g., figures 3 –35 and its related description); and

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description).

the well-known concept of transforming and usage of event transformer (e.g., figures 3 –35 and its related description).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., figures 3 –35 and its related description),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),

the second event filter has no knowledge of the first event (e.g., figures 3 –35 and its related description),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., figures 3 –35 and its related description), each of the plurality of event filters having an associated filter criteria (e.g., figures 3 –35 and its related description),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figures 3 –35 and its related description), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figures 3 –35 and its related description)

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figures 3 –35 and its related description),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., figures 3 –35 and its related description),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., figures 3 –35 and its related description),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figures 3 –35 and its related description),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figures 3 –35 and its related description) that receives the second event to identify an action to perform in response to the second event (e.g., figures 3 –35 and its related description),

the event action handler performs at least one action in response to the second event (e.g., figures 3 –35 and its related description).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figures 3 –35 and its related description).

29. Claims 1-29, are rejected under 35 U.S.C. 102(e) as being anticipated by Hinson et al. 2005/0044554 (Hereinafter Hinson4).

30. As per claims 1-29, Hinson4 teaches a method comprising:

a method comprising,

one or more computer-readable media having stored thereon a computer program that,

when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., figures 3 –35 and related description),

the first event filter having an associated filter criteria (e.g., figures 3 –35 and its related

description),

applying the filter criteria associated with the first event filter to the first event (e.g.,
figures 3 –35 and its related description),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., figures 3 –35 and its related
description); and

communicating the second event to a second event filter having an associated filter
criteria (e.g., figures 3 –35 and its related description),

the second event filter being associated with an event consumer wherein the event
consumer performs an action if the second event satisfies the filter criteria associated with the
second event filter (e.g., figures 3 –35 and its related description),

receiving a first event having a first format (e.g., figures 3 –35 and its related description
(),

processing the first event into a second event having a second format (e.g., figures 3 –35
and its related description),

Art Unit: 2154

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., figures 3 –35 and its related description),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),

an event processor to receive a first event and processing the first event into a second event (e.g., figures 3 –35 and its related description), the second event having a standard data format regardless of the first event data format (e.g., figures 3 –35 and its related description),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., figures 3 –35 and its related description),

a plurality of event consumers coupled to the plurality of event filters (e.g., figures 3 –35 and its related description) and the event consumers to perform an action if the second event satisfies the filter criteria applied by the event filters (e.g., figures 3 –35 and its related description),

receive a first event having a first data format (e.g., figures 3 –35 and its related description),

filter the first event using a first filter criteria (e.g., figures 3 –35 and its related description),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description), wherein the second data format includes an event header having a plurality of parameters (e.g., figures 3 –35

Art Unit: 2154

and its related description), and an event payload having a plurality of payload objects (e.g., figures 3 –35 and its related description); and

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., figures 3 –35 and its related description).

the well-known concept of transforming and usage of event transformer (e.g., figures 3 –35 and its related description).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., figures 3 –35 and its related description),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 3 –35 and its related description),

the second event filter has no knowledge of the first event (e.g., figures 3 –35 and its related description),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., figures 3 –35 and its related description), each of the plurality of event filters having an associated filter criteria (e.g., figures 3 –35 and its related description),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figures 3 –35 and its related description), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figures 3 –35 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figures 3 –35 and its related description)

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figures 3 –35 and its related description),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., figures 3 –35 and its related description),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., figures 3 –35 and its related description),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figures 3 –35 and its related description),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figures 3 –35 and its related description) that receives the second event to identify an action to perform in response to the second event (e.g., figures 3 –35 and its related description),

the event action handler performs at least one action in response to the second event (e.g., figures 3 –35 and its related description).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figures 3 –35 and its related description).

31. Claims 1-29, are rejected under 35 U.S.C. 102(e) as being anticipated by Jakobson et al. 6,766,368 (Hereinafter Jakobson).

32. As per claims 1-29, Jakobson teaches a method comprising:

a method comprising,

one or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to,

an apparatus comprising:

receiving a first event at a first event filter (e.g., figures 1-10 and its related description), the first event filter having an associated filter criteria (e.g., figures 1-10 and its related description),

applying the filter criteria associated with the first event filter to the first event (e.g., figures 1-10 and its related description),

if the first event satisfies the filter criteria associated with the first event filter, then:

processing the first event into a second event (e.g., figures 1-10 and its related description); and

communicating the second event to a second event filter having an associated filter criteria (e.g., figures 1-10 and its related description),

the second event filter being associated with an event consumer wherein the event consumer performs an action if the second event satisfies the filter criteria associated with the second event filter (e.g., figures 1-10 and its related description),

receiving a first event having a first format (e.g., figures 1-10 and its related description),

processing the first event into a second event having a second format (e.g., figures 1-10 and its related description),

generating an event header having a plurality of parameters, wherein the plurality of parameters are arranged in a standard data format (e.g., figures 1-10 and its related description),

generating an event payload having a plurality of payload objects wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 1-10 and its related description),

an event processor to receive a first event and processing the first event into a second event (e.g., figures 1-10 and its related description), the second event having a standard data format regardless of the first event data format (e.g., figures 1-10 and its related description),

a plurality of event filters coupled to the event processor, the event filters to apply filter criteria to the second event (e.g., figures 1-10 and its related description),

a plurality of event consumers coupled to the plurality of event filters (e.g., figures 1-10 and its related description) and the event consumers to perform an action if the second event

Art Unit: 2154

satisfies the filter criteria applied by the event filters (e.g., figures 1-10 and its related description),

receive a first event having a first data format (e.g., figures 1-10 and its related description),

filter the first event using a first filter criteria (e.g., figures 1-10 and its related description),

processing the first event into a second event having a second data format if the first event satisfies the first filter criteria (e.g., figures 1-10 and its related description), wherein the second data format includes an event header having a plurality of parameters (e.g., figures 1-10 and its related description), and an event payload having a plurality of payload objects (e.g., figures 1-10 and its related description); and

communicate the second event to an event action handler if the first event satisfies the first filter criteria (e.g., figures 1-10 and its related description).

the well-known concept of transforming and usage of event transformer (e.g., figures 1-10 and its related description).

the second event includes a header having a plurality of parameters wherein the event header has a standard data format regardless of event source (e.g., figures 1-10 and its related description),

the second event includes a payload including a plurality of payload objects, wherein the plurality of payload objects identify at least one action to perform in response to the event (e.g., figures 1-10 and its related description),

the second event filter has no knowledge of the first event (e.g., figures 1-10 and its related description),

communicating the second event to a second event filter further comprises communicating the second event to a plurality of event filters (e.g., figures 1-10 and its related description), each of the plurality of event filters having an associated filter criteria (e.g., figures 1-10 and its related description),

each of the plurality of event filters being associated with one of a plurality of event consumers (e.g., figures 1-10 and its related description), wherein each of the plurality of event consumers performs an action if the second event satisfies the filter criteria associated with the corresponding event filter (e.g., figures 1-10 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is logging the second event to a storage device (e.g., figures 1-10 and its related description).

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is forwarding the second event to a destination (e.g., figures 1-10 and its related description)

the action performed by the event consumer if the second event satisfies the filter criteria associated with the second event filter is generating an email message (e.g., figures 1-10 and its related description),

applying the plurality of parameters in the event header to a filter to determine whether the associated event meets criteria associated with the filter, the second event includes an event header having a plurality of parameters arranged in a standard data format, and wherein the

plurality of parameters in the event header are applied to the event filters to determine whether the associated event satisfies the filter criteria (e.g., figures 1-10 and its related description),

the plurality of parameters are arranged in a standard data format regardless of the first event source, applying the second event to an event filter having an associated filter criteria; and communicating the second event to an event consumer if the second event satisfies the filter criteria associated with the event filter (e.g., figures 1-10 and its related description),

the event processor operates independently of the event filters and independently of the event consumers (e.g., figures 1-10 and its related description),

the plurality of payload objects in the event payload are used by an event consumer (e.g., figures 1-10 and its related description) that receives the second event to identify an action to perform in response to the second event (e.g., figures 1-10 and its related description),

the event action handler performs at least one action in response to the second event (e.g., figures 1-10 and its related description).

wherein the filter criteria associated with the first event filter / the second filter includes an event type (e.g., figures 1-10 and its related description).

Conclusion

Examiner has cited particular columns and line numbers and/or paragraphs and/or sections and/or page numbers in the reference(s) as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing

Art Unit: 2154

responses, to fully consider the references in entirety, as potentially teaching, all or part of the claimed invention, as well as the context of the passage, as taught by the prior art or disclosed by the Examiner.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Haresh Patel whose telephone number is (571) 272-3973. The examiner can normally be reached on Monday, Tuesday, Thursday and Friday from 10:00 am to 8:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on (571) 272-3964. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Haresh Patel

April 17, 2006

Haresh Patel
[Signature]